สุทธิศักดิ์ พงศ์ธนาพานิช

Advanced BASIC Programming Technique

ตอน Tips and Tricks for Visual BASIC

แนะนำเทคนิคใหม่ ๆ ที่ช่วยเพิ่มความเร็วและขยายความสามารถของ VB/WIN โดยใช้ Windows API

Visual BASIC เป็นโปรแกรมที่สามารถใช้เขียนโปรแกรมบนวินโดวส์ได้โดยไม่ยุ่งยากมากนัก แต่โปรแกรมเมอร์ที่ใช้ VB/Win มักจะมีคำถามที่คล้ายกันก็คือ จะมีวิธีใดบ้างที่สามารถเพิ่มความเร็วและประสิทธิภาพของแอพพลิเคชันที่พัฒนาโดยใช้ VB/Win ดังนั้นสำหรับบทความตอนนี้จึงเป็นการแนะนำ trick ต่าง ๆ ที่เพิ่มความเร็วและขยายความสามารถของ VB/Win โดยการใช้ Windows API ซึ่งจะเป็นเส้นทางใหม่สำหรับนักพัฒนาแอพพลิเคชันโดยใช้ VB/Win และในตอนท้ายของบทความนี้ผมก็จะแนะนำฟังก์ชัน Windows API และเครื่องมือที่ช่วยสนับสนุนการทำงานหรือการพัฒนาโปรแกรมบน VB/Win เพื่อให้ผู้อ่านสามารถนำไปใช้ในการพัฒนาโปรแกรมด้วยตัวเอง

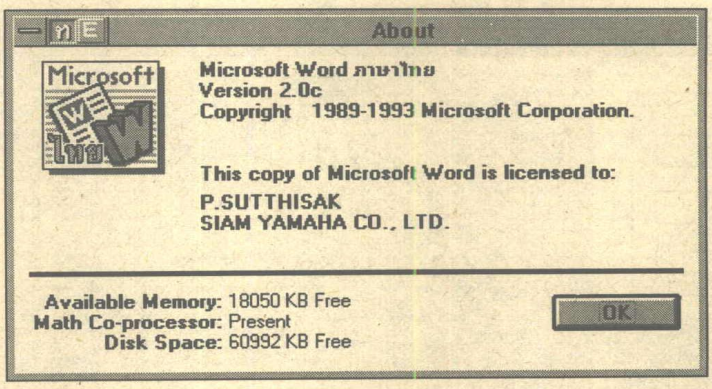
การเพิ่มความเร็วให้กับแอพพลิเคชัน

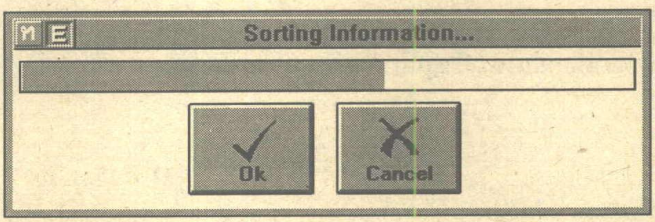
การเพิ่มความเร็วของแอพพลิเคชันนั้นสามารถทำได้โดยหลาย ๆ วิธี เช่น การเพิ่มความเร็วของฮาร์ดแวร์ แต่มันไม่ใช่วิธีที่ดีที่สุด เพราะแอพพลิเคชันที่ดีควรจะทำงานได้อย่างรวดเร็วอย่างเพียงพอบนเครื่องที่มีฮาร์ดแวร์ที่อยู่ในสภาพที่ดีพอรับได้ ดังนั้นหนทางที่ดีกว่าการเพิ่มเงินเพื่อเพิ่มประสิทธิภาพทางฮาร์ดแวร์ก็คือเทคนิคการจัดการด้านซอฟต์แวร์ โดยมีหัวข้อหลัก ๆ ดังนี้

- **การใช้ฟังก์ชัน Windows API**
- **การรวมฟอร์ม (form) และโมดูล (module)**
- **การเพิ่มความเร็วการระบายสกรีน**
- **การพรีโหลดฟอร์ม**

การใช้งานฟังก์ชัน Windows API เป็นวิธีที่ดีที่สุดในการขยายขอบเขตความสามารถให้กับแอพพลิเคชันที่พัฒนาบน VB/Win เพราะฟังก์ชันต่าง ๆ ก็มีมาให้อยู่แล้วภายในวินโดว์ (add-in) ดังนั้นราคาของฟังก์ชันเหล่านี้ก็คงจะขึ้นกับเวลาที่นักพัฒนาแอพพลิเคชันทั้งหลายจะใช้ในการศึกษาถึงความสามารถและวิธีการเข้าถึงฟังก์ชันเหล่านี้เท่านั้น

การรวมฟอร์มและโมดูลเป็นวิธีการอย่างหนึ่งที่มักจะถูกนักพัฒนาแอพพลิเคชันทั้งหลายมองข้ามไป การนำฟอร์มที่มีอยู่แล้วมาใช้ใหม่นั้นเป็นแนวความคิดที่ดีเพราะสามารถลดขนาดของโปรแกรมและลดเวลาในการประมวลผลของโปรแกรมลงได้อีกด้วย อย่างเช่นฟอร์มสำหรับ About ซึ่งโดยทั่วไปจะถูกใช้ในการแสดงชื่อของผู้พัฒนาแอพพลิเคชันและสถานะต่าง ๆ ของระบบในปัจจุบันดังในรูปที่ 1 ซึ่งฟอร์มของ About จะถูกใช้เพียงหน้าที่เดียวดังนั้นผู้อ่านก็สามารถที่จะทำการตบแต่งฟอร์มและขนาดเสียใหม่เพื่อให้สามารถนำฟอร์มที่มีอยู่แล้วนี้ไปใช้งานด้านอื่น ๆ ได้ด้วย เช่น ใช้เป็นฟอร์มสำหรับแสดงสถานะการทำงานของกระบวนการ





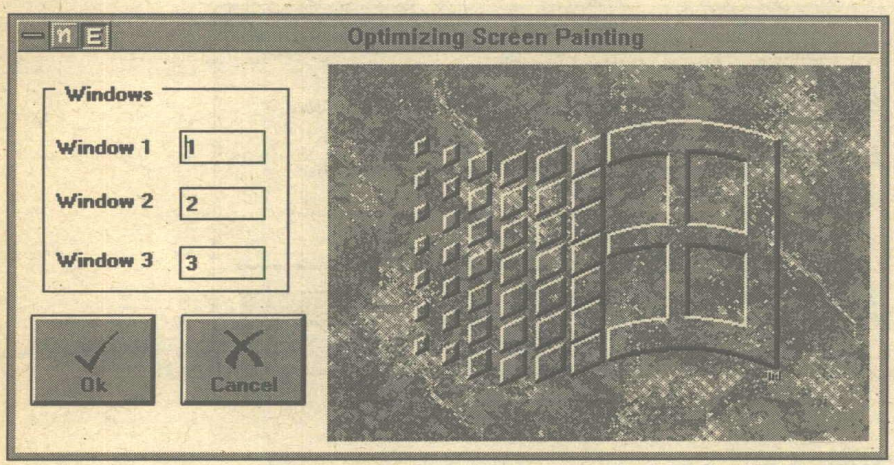
รูปที่ 1 แสดงการเปลี่ยน caption จาก Abont ให้เป็น Sorting Information

ต่าง ๆ โดยในรูปที่ 1 ผมได้เปลี่ยน caption จาก About มาเป็น Sorting Information... และได้ทำการซ่อนคอนโทรลส่วนที่ไม่เกี่ยวข้องโดยการกำหนดคุณสมบัติ visible = FALSE และแสดงคอนโทรลส่วนที่เกี่ยวข้องและปรับเปลี่ยนขนาดของฟอร์มพร้อมทั้งเปลี่ยนคุณสมบัติบางอย่างของฟอร์ม เช่น control menu เป็นต้น

ส่วนการรวมเอาฟังก์ชันหรือรูทีนที่ทำงานเกี่ยวข้องกันเอาไว้ในโมดูลเดียวกันก็เป็นแนวความคิดที่ดีเพราะโมดูลจะถูกเรียกสู่หน่วยความจำก็ต่อเมื่อ VB/Win เรียกใช้ฟังก์ชันหรือรูทีนที่มีอยู่ในโมดูลนั้น ๆ เช่น ถ้าหากเราเขียนฟังก์ชันเกี่ยวกับขบวนการพิมพ์ก็ควรจะแยกเก็บลงในโมดูลหนึ่งโดยสมมติว่าชื่อ PRINTING.BAS และฟังก์ชันอีกส่วนทำงานเกี่ยวกับการระบายสกรีนก็จะถูกแยกเก็บในอีกโมดูลที่ชื่อ PAINTING.BAS ดังนั้นเมื่อ VB/Win เรียกใช้ฟังก์ชันหรือรูทีนที่มีอยู่ในโมดูลเฉพาะโมดูลที่เก็บฟังก์ชันนั้น ๆ เท่านั้นที่จะถูกเรียกลงสู่หน่วยความจำ ดังนั้นจึงทำให้ VB/Win สามารถทำงานกับโมดูลนั้น ๆ ได้ง่ายขึ้น และนอกจากนี้ยังทำให้การแก้ไขหรือดัดแปลงโปรแกรมสามารถทำได้ง่ายขึ้นเพราะฟังก์ชันที่ทำงานเกี่ยวข้องกันได้ถูกรวมกลุ่มเอาไว้ในโมดูลเดียวกันอยู่แล้ว

การเพิ่มความเร็วการระบายสกรีนก็เป็นอีกวิธีที่จะเพิ่มความเร็วของแอพพลิเคชันได้อีกด้านหนึ่ง ซึ่งมีเพียงวิธีเดียวที่สามารถทำได้ก็โดยการปรับเปลี่ยนลำดับของคอนโทรลที่จะแสดงบนฟอร์มนั้น ๆ ซึ่งสามารถแสดงได้ง่าย ๆ ดังนี้

ในฟอร์มตัวอย่างในรูปที่ 2 ผมกำหนดให้ลำดับของคอนโทรลที่จะแสดงผลบนฟอร์มมีดังรูปที่ 3



รูปที่ 2 แบบฟอร์มที่แสดงจากลำดับคอนโทรลในรูปที่ 3

```
"Ok" Button
"Cancel" Button
"Windows" Frame
Text "1"
Text "2"
Text "3"
Lable "Window 1"
Lable "Window 2"
Lable "Window 3"
Image "Windows"
```

รูปที่ 3 แสดงลำดับของคอนโทรลเพื่อให้แสดงผลในรูปที่ 2

และโปรแกรมในรูปที่ 4 นี้เป็นเพียงตัวอย่างส่วนหนึ่งที่ของโปรแกรมจากฟอร์มในรูปที่ 2 โดยที่ส่วนของคุณสมบัติของแต่ละคอนโทรลผมจะเลือกแสดงเฉพาะ caption เพื่อให้ง่ายต่อการสังเกตุและเปรียบเทียบกับฟอร์มที่ได้

ดังนั้นถ้าหากผู้อ่านต้องการแก้ไขลำดับการแสดงผลให้ตรงตามที่ต้องการก็ให้แก้ไข text file ของโปรแกรมในรูปที่ 4 โดยการเรียงลำดับของคอนโทรลเสียใหม่ดังโปรแกรมในรูปที่ 5

จะเห็นว่าลำดับการระบายสกรีนได้ถูกเปลี่ยนให้ตรงตาม Tab Order ตามที่เราต้องการ แต่เนื่องจากฟอร์มตัวอย่างที่แสดงให้ดูรูปที่ 5 นี้ไม่ซับซ้อนผู้อ่านจึงอาจจะไม่สามารถเห็นผลของความแตกต่างมากนัก แต่ถ้าหากฟอร์มที่เราทำการแก้ไขลำดับการระบายสกรีนให้ตรงกับ Tab Order มีความซับซ้อนมีคอนโทรลมาก ๆ ก็จะทำให้เราสามารถเห็นผลของความแตกต่างได้มากยิ่งขึ้น สำหรับการแก้ไข text file ของฟอร์มนั้นให้กระทำด้วยความระวัง เพราะถ้าหากผู้อ่านลบบรรทัดใดบรรทัดหนึ่งเช่น End หรือ Begin เป็นต้นออกไป ผลที่ได้อาจจะทำให้ฟอร์มนั้นไม่สามารถถูกโหลดโดย VB/Win ได้เช่นกัน

การพรีโหลดฟอร์ม หมายถึงการโหลดฟอร์มลงในหน่วยความจำก่อนและซ่อนมันเอาไว้จนกว่าจะต้องการแสดงฟอร์มนั้น ๆ เป็นวิธีที่ง่ายที่สุดที่สามารถเพิ่มความเร็วให้กับแอพพลิเคชันแทนการ load และ unload แต่ข้อเสียของวิธีนี้ก็คือ การโหลดฟอร์มลงในหน่วยความจำหลาย ๆ ฟอร์มจะทำให้แอพพลิเคชันใช้หน่วยความจำมาก ดังนั้นถ้าหากฟอร์มมีขนาดใหญ่มาก ๆ วิธีนี้ก็ไม่ใช่แนวความคิดที่ดีนักไม่ใช่หรือ?

การเพิ่มพื้นที่ว่างของ resource

หลังจากที่เราได้รู้ถึงวิธีการเพิ่มความเร็วให้กับแอพพลิเคชันแล้วต่อไปก็จะพูดถึงการเพิ่มพื้นที่ว่างให้กับทรัพยากรสำหรับแอพพลิเคชันของ VB (resource) นั่นก็หมายความว่าแอพพลิเคชันมีพื้นที่ว่างของสแต็กเพียงพอสำหรับการทำงานที่ดีกว่า เพราะแอพพลิเคชันที่มีขนาดใหญ่จะทำงานได้ไม่ดีนักหรือบางครั้งอาจจะไม่สามารถทำงานต่อไปได้อีกถ้าหากมีพื้นที่ว่างของสแต็กไม่เพียงพอสำหรับการทำงาน ซึ่งการเพิ่มทรัพยากรที่สามารถนำไปใช้งานได้มีอยู่หลายวิธีดังนี้

- **การใช้งานฟังก์ชัน Windows API**
- **ใช้คอนโทรล image แทน picture สำหรับการแสดงผลกราฟิก**
- **ลดปริมาณการใช้คอนโทรลชนิด 3 มิติ (three dimensional control)**

```
Begin Form Form2
   Caption          =   "Optimizing Screen
Painting"
   Begin SSCommand Command3D2
      Caption       =   "Cancel"
   End
   Begin SSCommand Command3D1
      Caption       =   "Ok"
   End
   Begin Frame Frame1
      Caption       =   " Windows "
      Begin TextBox Text3
         Text       =   "3"
      End
      Begin TextBox Text2
         Text       =   "2"
      End
      Begin TextBox Text1
         Text       =   "1"
      End
      Begin Label Label3
         Caption    =   "Window 3"
      End
      Begin Label Label2
         Caption    =   "Window 2"
      End
      Begin Label Label1
         Caption    =   "Window 1"
      End
   End
   Begin Image Image1
   End
End
```

รูปที่ 4 โปรแกรมแบบฟอร์มในรูปที่ 2 ที่ยังไม่ถูก Optimize

```
Begin Form Form2
   Caption          =   "Optimizing Screen
Painting"
   Begin Frame Frame1
      Caption       =   " Windows "
      Begin Label Label1
         Caption    =   "Window 1"
      End
      Begin TextBox Text1
         Text       =   "1"
      End
      Begin Label Label2
         Caption    =   "Window 2"
      End
      Begin TextBox Text2
         Text       =   "2"
      End
      Begin Label Label3
         Caption    =   "Window 3"
      End
      Begin TextBox Text3
         Text       =   "3"
      End
   End
   Begin SSCommand Command3D1
      Caption       =   "Ok"
   End
   Begin SSCommand Command3D2
      Caption       =   "Cancel"
   End
   Begin Image Image1
   End
End
```

รูปที่ 5 โปรแกรมแบบฟอร์มในรูปที่ 2 ที่ถูก Optimize เรียงลำดับคอนโทรลใหม่

สำหรับการใช้งานฟังก์ชัน window API นั้นได้กล่าวถึงมาแล้วในช่วงต้นของบทความจึงไม่ขอกล่าวซ้ำอีก image เป็นคอนโทรลที่ใช้ทรัพยากรน้อยกว่าและสามารถระบายได้เร็วกว่า picture แต่ไม่ว่าจะเป็นด้านคุณสมบัติ (property), เหตุการณ์ (event) หรือ วิธีการ (method) ที่มีอยู่ใน image ก็จะเป็นส่วนหนึ่งของ picture ดังรายละเอียดในรูปที่ 6

ในอดีตที่ผ่านมาการสร้าง GUI ต่าง ๆ มักจะใช้คอนโทรลชนิด 3 มิติเพื่อเพิ่มความเหมือนจริงให้มากยิ่งขึ้น แต่ปัญหาของคอนโทรลชนิด 3 มิติก็คือมันต้องใช้ทรัพยากรสำหรับการแสดงผลของระบบอย่างมากมายเลยทีเดียว ดังนั้นวิธีแก้ไขก็คือการใช้ไฟล์ icon (.ICO) แทน โดยที่เราจะต้องสร้างรูปโดยการใช้วิธีการให้สีหรือเงาให้เหมือนกับภาพ 3 มิติจริง ๆ แล้วจัดเก็บลงในไฟล์ icon ซึ่งวิธีการสร้างก็สามารถทำได้ง่ายโดยใช้โปรแกรม IconWorks ที่มีมาให้ฟรีพร้อมกับโปรแกรม VB/Win หรือถ้าหากผู้อ่านไม่ต้องการสร้างไฟล์ icon ก็สามารถใช้วิธีอื่นโดยใช้โปรแกรมประเภทจับภาพจากหน้าจอ (screen capture) แล้วเก็บภาพที่ได้เป็นไลบรารี่สำหรับนำมาใช้ต่อไป

การเพิ่มพื้นที่ว่างของหน่วยความจำ RAM

หลังจากที่รู้ถึง trick สำหรับการเพิ่มความเร็ว การเพิ่มทรัพยากรที่สามารถนำไปใช้ประโยชน์ได้ ยังมีอีกสิ่งที่มีความสำคัญมากสำหรับแอพพลิเคชันที่ทำงานบนวินโดว์ก็คือ หน่วยความจำ (RAM) การเพิ่มพื้นที่ว่างให้กับหน่วยความจำสามารถทำได้หลายวิธี ดังนี้

Image Property

BorderStyle	Enabled	Left	Picture	Top	
DragIcon	Height	MousePointer	Stretch	Visible	
DragMode	Index	Name	Tag	Width	

Picture Property

*AutoRedraw	*FontItalic	*ScaleTop	Height	MousePointer	
*ClipControl	*FontName	*ScaleWidth	HelpContextID	Name	
*CurrentX	*FontSize	Align	hWnd	Parent	
*CurrentY	*FontStrikethru	AutoSize	Image	Picture	
*DrawMode	*FontTransparent	BackColor	Index	TabIndex	
*DrawStyle	*FontUnderline	BorderStyle	Left	TabStop	
*DrawWidth	*ForeColor	DragIcon	LinkItem	Tag	
*FillColor	*ScaleHeight	DragMode	LinkMode	Top	
*FillStyle	*ScaleLeft	Enabled	LinkTimeout	Visible	
*FontBold	*ScaleMode	hDC	LinkTopic	Width	

Image Event

Click	DragDrop	MouseDown	MouseUp	
DblClick	DragOver	MouseMove		

Picture Event

Change	DragOver	KeyUp	LinkOpen	MouseUp
Click	GotFocus	LinkClose	LostFocus	Paint
DblClick	KeyDown	LinkError	MouseDown	Resize
DragDrop	KeyPress	LinkNotify	MouseMove	

Image Method

Drag	Move	Refresh	Zorder

Picture Method

*Circle	*Point	*Scale	Drag	LinkRequest	Refresh
*Cls	*Print	*TextHeight	LinkExecute	LinkSend	
SetFocus					
*Line	*Pset	*TextWidth	LinkPoke	Move	Zorder

* แสดงถึงวิธีการทางด้านกราฟิก

รูปที่ 6

- **สร้างตัวแปรอะเรย์ชนิดไดนามิกส์**
- **unload คอนโทรลหรือฟอร์ม**
- **กำหนดอายุ (lifetime) การปฏิบัติงานของตัวแปร**

การเขียนโปรแกรมที่มีการใช้งานตัวแปรอะเรย์ควรจะกำหนดให้เป็นชนิดไดนามิกส์ เพื่อให้การใช้หน่วยความจำเป็นไปตามปริมาณที่ต้องการจริง ๆ เช่น สมมติว่าโปรแกรมใช้ตัวแปรอะเรย์ Student สำหรับโรงเรียนแห่งหนึ่งที่มีจำนวนนักเรียนมากที่สุด 1,200 คน ถ้าหากผู้อ่านจองหน่วยความจำให้กับตัวแปรอะเรย์เป็นแบบสแตติกดังนี้

```
DIM Student (100)
```

ดังนั้นถ้าหากปีการศึกษาใดที่มีจำนวนนักเรียนไม่ครบ 1,200 คน สมาชิกส่วนที่เกินก็จะไม่ถูกใช้งานแต่หน่วยความจำส่วนนี้จะไม่สามารถนำไปใช้ประโยชน์โดยตัวแปรอื่น ๆ ได้ แต่ถ้าหากผู้อ่านกำหนดตัวแปรอะเรย์เป็นชนิดไดนามิกส์ดังนี้

```
DIM Student ( )
```

แล้วทำการเพิ่มหรือลด (โดยใช้ประโยค Preserve) จำนวนสมาชิกให้ครบตามจำนวนนักเรียนที่มีอยู่จริง ๆ ก็จะทำให้เกิดการใช้หน่วยความจำตรงตามที่ต้องการจริง ๆ ตลอดเวลา

ต่อมาก็เป็นการ unload ฟอร์มหรือคอนโทรลส่วนที่ไม่ถูกใช้งานอีกต่อไปก็จะเป็นการเพิ่มหน่วยความจำได้อย่างมากทีเดียว เช่น คอนโทรลที่ถูกสร้างขึ้นเป็นอะเรย์โดยประโยค Load ก็ให้ทำการยกเลิกส่วนที่ไม่จำเป็นต้องใช้งานด้วยประโยค Unload นั่นเอง

สุดท้ายก็เป็นการกำหนดอายุของตัวแปร ซึ่งในที่นี้หมายถึงการใช้หรือ

การนำตัวแปรที่มีอยู่แล้วกับมาใช้ใหม่อีกครั้ง ซึ่งวิธีที่ง่ายสำหรับการควบคุมการใช้ตัวแปรก็โดยการใช้ประโยค Option Explicit เพราะเมื่อใช้ประโยคนี้ตัวแปรทุกตัวที่ใช้งานในโปรแกรมจะต้องถูกประกาศเสียก่อนด้วยประโยค Dim, ReDim หรือ Global ไม่เช่นนั้นถ้าหากเรารันโปรแกรมก็จะเกิดข้อผิดพลาดขึ้นมา ดังนั้นเมื่อตัวแปรทุกตัวถูกประกาศก็ทำให้เราสามารถควบคุมสถานะและขอบเขตที่ตัวแปรนั้นจะสามารถใช้งานได้โดยง่าย

ฟังก์ชัน windows API

สิ่งหนึ่งที่มีความสำคัญมากในการพัฒนาแอพพลิเคชันภายใต้วินโดว์ก็คือการใช้ฟังก์ชัน window API เพราะนอกจากช่วยเพิ่มความเร็วให้กับแอพพลิเคชันแล้ว ยังช่วยประหยัดเงินสำหรับการซื้อ custom control จากบริษัทผู้พัฒนาเครื่องมือสำหรับ VB/Win ได้อีกด้วย ซึ่งฟังก์ชัน API กว่า 700 ฟังก์ชันได้ถูกแบ่งออกเป็นกลุ่ม ๆ ได้ 4 กลุ่มสำหรับโปรแกรมเมอร์ด้วย VB/Win ดังนี้

- **ฟังก์ชัน API ที่ตรงกันกับคุณสมบัติที่มีใน VB/Win**
- **ฟังก์ชัน API ที่ไม่สามารถนำมาใช้กับ VB/Win**
- **ฟังก์ชัน API ที่สามารถนำมาใช้และมีประโยชน์สำหรับนักพัฒนาแอพพลิเคชันด้วย VB/Win**
- **ฟังก์ชัน API ที่ไม่สามารถเรียกจาก VB/Win**

และสำหรับผู้ที่สนใจเกี่ยวกับการใช้งานฟังก์ชัน API ร่วมกับ VB/Win ก็ขอแนะนำหนังสือที่เขียนเกี่ยวกับเรื่องนี้เอาไว้ดีมากเล่มหนึ่งชื่อ Visual BASIC Programmers Guide to the Windows API ต่อไปนี้เป็นส่วนหนึ่งของฟังก์ชันที่สามารถช่วยในการเขียนโปรแกรมตามที่กล่าวในข้างต้นที่ผ่านมาพอสังเขปดังนี้

```
Declare Function FindWindow Lib "User" (ByVal
lpClassName As Any, ByVal
    lpWindowName As Any)
Declare Function SendMessage Lib "User"
(ByVal hWnd As Integer, ByVal
    wMsg As Integer, ByVal wParam As Integer,
lParam As Any)
Declare Function PostMessage Lib "User"
(ByVal hWnd As Integer, ByVal
    wMsg As Integer, ByVal wParam As Integer,
lParam As Any)
```

รูปที่ 7

```
Declare Function SetSysModalWindow Lib "User"
(ByVal hWnd As Integer)
```

รูปที่ 8

ตัวอย่างการส่ง message ระหว่างแอพพลิเคชัน

VB/Win ช่วยให้นักพัฒนาแอพพลิเคชันสามารถเขียนโปรแกรมภายใต้วินโดว์ได้โดยไม่จำเป็นต้องมีพื้นฐานเกี่ยวกับการเขียนโปรแกรมวินโดว์หรือการทำงานของวินโดว์ แต่ด้วยข้อดีนี้เองทำให้เกิดข้อเสียประการหนึ่งก็คือเราไม่สามารถที่จะสื่อสารระหว่างแอพพลิเคชันที่เขียนโดย VB/Win ซึ่งวิธีแก้ไขข้อบกพร่องนี้ก็โดยใช้ฟังก์ชัน API ช่วยในการส่ง message หรือข้อความไปมาระหว่างแอพพลิเคชัน โดยที่ฟังก์ชันมีการประกาศดังรูปที่ 7

ตัวอย่างการควบคุมเดสก์ท้อป

วิธีที่จะกำหนดให้แอพพลิเคชันไม่ถูกเปลี่ยนไปยังแอพพลิเคชันตัวอื่น ๆ ได้ก็สามารถทำได้โดยการกำหนดให้แอพพลิเคชันเป็น System Model โดยใช้ฟังก์ชัน API แต่ควรใช้ด้วยความระวัง สำหรับการประกาศฟังก์ชันดังรูปที่ 8

ตัวอย่างการควบคุมพื้นที่การเคลื่อนที่ของเคอร์เซอร์

การกำหนดพื้นที่สี่เหลี่ยมสำหรับการเคลื่อนที่ของเคอร์เซอร์เราเรียกว่า clipping สามารถทำได้โดยการกำหนดตำแหน่งโคออร์ดิเนต top, bottom, left และ right ให้กับตัวแปร IpRect ของฟังก์ชันดังรูปที่ 9

และถ้าหากต้องการยกเลิกการจำกัดพื้นที่การเคลื่อนที่ก็สามารถทำได้โดยการกำหนดค่า 0& ให้กับฟังก์ชันดังรูปที่ 10

ตัวอย่างการแก้ไขเมนูระบบ

ผู้อ่านสามารถที่จะยกเลิก Control box ของฟอร์มใน VB/Win ได้โดยง่าย โดยการกำหนดให้มีคุณสมบัติเป็น False แต่ถ้าหากต้องการให้แสดงเฉพาะบางรายการก็สามารถทำได้โดยผ่านทางฟังก์ชัน API ดังรูปที่ 11

ตัวอย่างการซ่อนแอพพลิเคชันอื่น

ในบางครั้งผู้อ่านอาจจะต้องการค้นหาตำแหน่งของแอพพลิเคชันอื่นแล้วซ่อนมันเอาไว้ก่อน ก็สามารถทำได้โดยใช้ฟังก์ชันทั้งสอง ดังรูปที่ 12

ตัวอย่างการขอข้อมูลจากระบบ

แอพพลิเคชันส่วนใหญ่จะมีคำสั่ง About... ในเมนู Help หรืออื่น ๆ ที่ใช้ในการแสดงข้อมูลของแอพพลิเคชันและชื่อของนักพัฒนาแอพพลิเคชันนั้น และนอกจากนี้ยังบอกถึงข้อมูลอื่น ๆ ของระบบ เช่น หน่วยความจำที่ว่าง, สถานะโปรเซสเซอร์ร่วม, เวอร์ชันของวินโดว์, พื้นที่ว่างของดิสก์ หรืออื่น ๆ สามารถทำได้โดยใช้ฟังก์ชัน ดังรูปที่ 13

สุดท้ายก่อนจบบทความก็ขอแนะนำเครื่องมือที่ช่วยลดเวลาการเขียนโปรแกรมและเพิ่มความสามารถในการใช้โปรแกรม VB/Win ได้สูงขึ้น ก็โดยการใช้ custom control จากนักพัฒนาเครื่องมือสำหรับ VB/Win (Third-Party VB/Win Developer) ดังต่อไปนี้เป็นเพียงตัวอย่างเครื่องมือที่เป็นที่นิยมใช้กันมากเท่านั้น

Message Blaster เป็นเครื่องมือตัวหนึ่งที่เป็น subclass ของ VB/Win ซึ่งช่วยในการสื่อสารระหว่างแอพพลิเคชันที่พัฒนาขึ้นมากับแอพพลิเคชันตัวอื่น ๆ โดยมีหลักการทำงานคร่าว ๆ ดังนี้เมื่อมีการส่งคำสั่ง message มายังแอพพลิเคชันของ VB/Win คำสั่ง

```
Declare Function ClipCursor Lib "User"
(lpRect As Any)
```

รูปที่ 9

```
Declare Function ClipCursor ByVal 0&
```

รูปที่ 10

```
Declare Function GetSystemMenu% Lib "User"
(ByVal hWnd%, ByVal bRevert%)
Declare Function ModifyMenu Lib "User" (ByVal
hMenu%, ByVal nPosition%
    ,ByVal wFlags%, ByVal wIDNewItem%, ByVal
lpString.As Any)
```

รูปที่ 11

```
Declare Function FindWindow Lib "User" (ByVal
lpClassName As Any, ByVal
    lpWindowName As Any)
Declare Function ShowWindow Lib "User" (ByVal
hWnd As Integer  ByVal
    nCmdShow As Integer)
```

รูปที่ 12

```
Declare Function GetFreeSpace Lib "Kernel"
(ByVal wFlags%) As Long
Declare Function GetWinFlags Lib "Kernel" ()
As Long
Declare Function GetVersion Lib "Kernel" ()
As Long
```

รูปที่ 13

ดังกล่าวก็จะถูกรับเอาไว้และตีความโดย Message Blaster ดังนั้นนอกจากคำสั่ง message ธรรมดาแล้วเรายังสามารถนิยามคำสั่งประเภท custom message ได้อีกด้วย ซึ่งเครื่องมือนี้ได้ถูกตีพิมพ์อธิบายไว้ใน Microsoft Developer Network หรือ MSDN ซึ่งสามารถติดต่อได้โดยผ่านทาง CompuServe (go msdnlib) หรือติดต่อโดยตรงได้ตามที่อยู่ข้างล่าง

Microsoft Developer Network
One Microsoft Way
Redmond, WA 98052-6399
(Tel. 800-759-5474)

VB Assist เป็นโปรแกรมที่ทำงานร่วมกับ VB/Win ซึ่งสามารถช่วยให้การเขียนโปรแกรมและการจัดการคอนโทรล คุณสมบัติ ภายในฟอร์มได้อย่างง่ายดายมากขึ้น เครื่องมือที่มีสามารถแบ่งได้ดังนี้

- การจัดการคอนโทรลดังนี้ alignment, array management, tab order นอกจากนี้ยังกำหนดให้หลาย ๆ คอนโทรลที่มีคุณสมบัติเหมือนกันสามารถเปลี่ยนแปลงไปพร้อม ๆ กันได้อีกด้วย
- การจัดการคุณสมบัติและกำหนดวิธีการจัดการเช่น การกำหนดเซตของคุณสมบัติ, การกำหนดการซิงโครไนซ์

ของคุณสมบัติ (template) หรือการขอดูภาพสำหรับ image หรือ picture เป็นต้น

● การช่วยงานด้านฐานข้อมูล VB Assist มีเอดิเตอร์ที่ช่วยให้การจัดการ text file หรือไฟล์ .FRM, .BAS ของ VB/Win ได้เป็นอย่างดี แต่ข้อเสียของเอดิเตอร์ตัวนี้ก็คือมันไม่สามารถที่จะแก้ไขไฟล์ที่มีขนาดโตกว่า 64 กิโลไบต์

● เครื่องมืออื่น ๆ ประกอบด้วย code assistant ช่วยในการจัดการโค้ดและฟอร์มในรูปแบบของ text editor ได้อย่างดีทีเดียว และ scrolling form ที่ช่วยในการสร้างฟอร์มที่มีขนาดใหญ่กว่าจอภาพและสามารถเลื่อนขึ้นหรือลงได้ไม่ว่าจะอยู่ในขณะออกแบบหรือรันโปรแกรม ผู้สนใจติดต่อได้ที่

: Sheridan Software Systems
65 Maxess Road
Melville, NY 11747 (Tel 516-753-0985, BBS 516-753-5452, Fax 516-753-3661)

VB Compress สำหรับนักพัฒนาแอพพลิเคชันที่จำเป็นต้องเขียนโปรแกรมขนาดโตมาก ๆ ไม่ควรพลาดเครื่องมือตัวนี้ เพราะมันช่วยในการสร้างไฟล์ .EXE ให้มีขนาดเล็กลงและให้สามารถทำงานได้เร็วขึ้นเท่าที่จะเป็นไปได้ ซึ่งซอฟต์แวร์ตัวนี้มีอยู่ด้วยกัน 2 เอดิชัน คือ Standard Editor และ Professional Editor แต่เพื่อให้สามารถใช้งานได้สูงสุดควรเลือกอย่างหลังจะดีกว่า ติดต่อได้ที่

: Ben Whipple
Whippleware
20 Cedar Street
Charlestown, MA 02129 (Tel 800-241-8727, Fax 617-241-8496)

TCP/IP for Visual BASIC เป็นเครื่องมือช่วยในการพัฒนาแอพพลิเคชันแบบเน็ตเวอร์กด้วย VB/Win โดยมีคุณสมบัติพื้นฐานดังนี้

● ต้องการหน่วยความจำหลักเพียง 64 กิโลไบต์ และมีไฟล์ไลบรารี่ .DLL

● สนับสนุนทั้ง Ethernet, Token ring, FDDI และ SLIP

● มีไฟล์ไลบรารี .DLL สำหรับ FTP, SMTP และ SNMP

● Berkeley 4.3 socket interface

ติดต่อได้ที่

: NetManage Inc.
20823 Stevens Creek Blvd.
Cupertino, CA 95014 (Tel 408-973-7171, Fax 408-257-6405)

Agility/VB เป็นเครื่องมือสำหรับนักพัฒนาโปรแกรมประกอบฐานข้อมูล (database application) ซึ่งผู้อ่านสามารถที่จะเขียนโปรแกรมจัดการฐานข้อมูลได้ในเวลาที่ไม่มากนัก นอกจากนั้นยังสามารถที่จะนำข้อมูลมาออกเป็นแบบฟอร์มรายงานผลได้อย่างสวยงามและง่ายดายโดยใช้เครื่องมือที่ชื่อ Crystal Report ซึ่งคุณสมบัติโดยทั่วไปมีดังนี้

● คำสั่งสำหรับการเขียนโปรแกรมทางด้านฐานข้อมูล

● จัดการกับไฟล์ dBASE พร้อมทั้ง memo และ index

● สามารถจัดสร้างไฟล์ native ที่มีฟิลด์ประเภทตัวแปรไม่จำกัดความยาวได้

● build-in QBE

● on-line help และโค้ดตัวอย่าง

ติดต่อได้ที่

: Apex Software Corporation
4516 Henry Street
Pittsburge, PA 15213 (Tel 412-681-4343, Fax 412-681-4384)

CodeBasic เป็นโปรแกรมประเภทสนับสนุนการจัดการไฟล์ด้านฐานข้อมูลด้วย VB/Win สำหรับเวอร์ชัน 1.1 ยังสนับสนุน VB/Dos อีกด้วย โดยมีคุณสมบัติทั่วไปดังนี้

● สนับสนุนการเข้าถึงและสร้างไฟล์ dBASE III/IV, FoxPro, Clipper

● เพิ่มความเร็วในการค้นหาในฐานข้อมูลโดยการใช้อินเด็กซ์ super-fast FoxPro

● สามารถใช้งานแบบ multi-user

● มีลักษณะการใช้คล้ายฟังก์ชันของ dBASE

ติดต่อได้ที่

: Sequiter Software Inc.
#209, 9644-54 Avenue
Edmonton, AB, Canada T6E-5V1
(Tel 403-437-2410, Fax 403-436-2999)

ในครั้งต่อไปผมจะเสนอตัวอย่างการเรียกใช้ฟังก์ชัน API ทั้งที่กล่าวมาข้างต้นและฟังก์ชันอื่น ๆ ที่มีประโยชน์ทางด้านกราฟิก เพื่อให้ผู้อ่านสามารถนำไปประยุกต์ใช้กับแอพพลิเคชันของตัวเองได้อย่างเต็มที่ และนอกจากนี้ยังมีเทคนิคพิเศษเกี่ยวกับการประกาศฟังก์ชัน API อย่างปลอดภัย (type-safe declaration) เพื่อให้การใช้งานฟังก์ชันเป็นไปอย่างปลอดภัยที่สุดเท่าที่จะเป็นไปได้ แล้วพบกันในตอนต่อไปครับ